



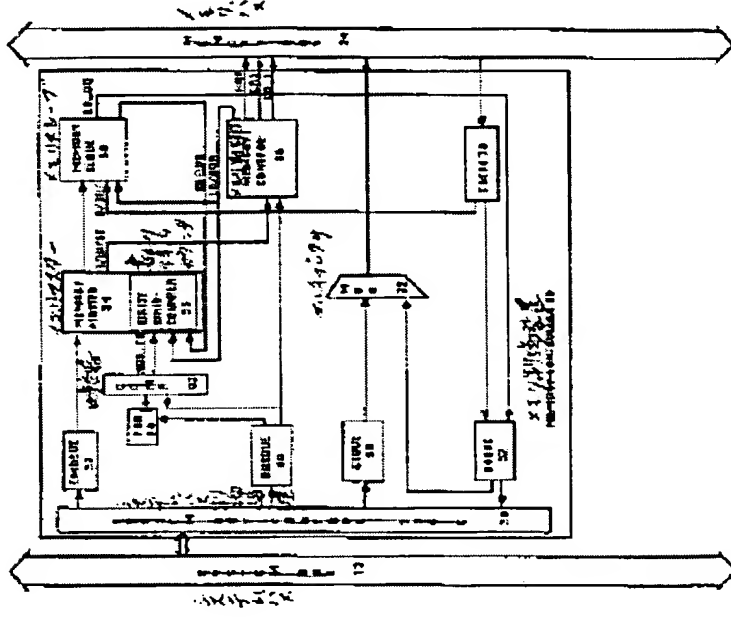
DATA PROCESSING SYSTEM AND METHOD FOR PREVENTING OVERWRITE

Patent number: JP6103153
 Publication date: 1994-04-15
 Inventor: GIRUMAN CHIESUREI
 Applicant: SUN MICROSYSTEMS INC
 Classification:
 - International: G06F12/00
 - european: G06F9/38D4
 Application number: JP19920353719 19921215
 Priority number(s): US19910809667 19911218

Also published as:
 EP0547769 (A1)
 EP0547769 (B1)

Abstract of JP6103153

PURPOSE: To prevent a problem of overwrite and to execute write duplication.
CONSTITUTION: Since a memory controller 16 constitutes a data read pipeline and a data write pipeline independently of each other, a write operation can overlap a preceding read operation. A limited number of write operations are selectively delayed; and only when a write operation accesses the same memory address as a preceding read operation, this write operation is delayed. This limited number of write operations are delayed to practically maintain advantages of write duplication while preventing a problem of overwrite. In an alternative example, an error write back operation related to a read operation is selectively suppressed if a write operation for the same address as the read operation follows this read operation.



Data supplied from the **esp@cenet** database - Worldwide

Ref-6

(19)日本国特許庁(J P)

(12) 公開特許公報(A)

(11)特許出願公開番号

特開平6-103153

(43)公開日 平成6年(1994)4月15日

(51)Int.Cl.⁵

G 0 6 F 12/00

識別記号

5 6 3

庁内整理番号

9366-5B

F I

技術表示箇所

審査請求 未請求 請求項の数3(全10頁)

(21)出願番号 特願平4-353719

(22)出願日 平成4年(1992)12月15日

(31)優先権主張番号 8 0 9 6 6 7

(32)優先日 1991年12月18日

(33)優先権主張国 米国(U S)

(71)出願人 591064003

サン・マイクロシステムズ・インコーポレーテッド

SUN MICROSYSTEMS, INCORPORATED

アメリカ合衆国 94043 カリフォルニア州・マウンテンビュー・ガルシア アヴェニュー・2550

(72)発明者 ギルマン・チェスレイ

アメリカ合衆国 95060 カリフォルニア州・サンタクルーズ・ペルトン アヴェニュー・1010

(74)代理人 弁理士 山川 政樹

(54)【発明の名称】 データ処理システム及び重ね書きを防止する方法

(57)【要約】

【目的】 重ね書きの問題を防止して書き込み重複を実行できるようにする。

【構成】 メモリ制御装置はデータ読取りパイプラインと、データ書き込みパイプラインとを別個に構成するので、書き込み動作は先行する読取り動作と重複することができる。本発明はいくつかの限定された数の書き込み動作を選択的に遅延させ、書き込み動作が先行する読取り動作と同じメモリアドレスを指向する場合にのみ、その書き込み動作を遅延させる。この限定された数の書き込み動作を遅延させることにより、本発明は重ね書きの問題を防止しつつ書き込み重複の利点を実質的に維持する。本発明の代替実施例は、読取り動作の後に同一のアドレスに対する書き込み動作が続いている場合に、読取り動作と関連する誤りライトバック動作を選択的に抑制する。

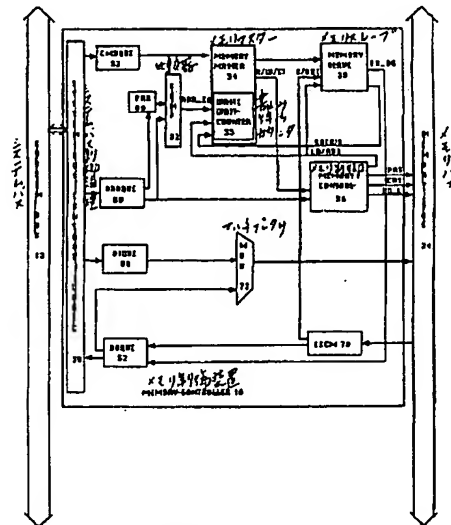


FIGURE 2

【特許請求の範囲】

【請求項1】 重ね書き防止を伴う書き込み重複を有するデータ処理システムにおいて、プロセッサと；データを記憶し且つ検索するメモリ手段と；前記プロセッサ及び前記メモリ手段に結合し、書き込み動作と、読み込み動作とを含むメモリ動作を制御し、所定の条件が満たされない限り、先行する読取り動作の完了に先立って書き込み動作を開始させるメモリ制御装置手段とを具備するデータ処理システム。

【請求項2】 重ね書き防止を伴う書き込み重複を有するデータ処理システムにおいて、プロセッサと；データを記憶し且つ検索するメモリ手段と；前記プロセッサ及び前記メモリ手段に結合し、読取り動作、書き込み動作及び誤りライトバック動作のためにメモリアドレスと、データと、制御信号とを供給するメモリ制御装置手段とを具備し、前記メモリ制御装置手段は、先行する読取り動作の完了に先立って書き込み動作を開始させる手段と、重ね書きが起こるのを防止する手段とを含むデータ処理システム。

【請求項3】 プロセッサと、メモリと、メモリアドレスを指向する読取り動作、書き込み動作及び誤りライトバック動作を含むメモリ動作を制御するメモリ制御装置とを有する書き込み重複を伴うデータ処理システムで重ね書きを防止する方法において、書き込み動作に関わるメモリアドレスを先行する読取り動作に関わるメモリアドレスと比較する過程と；重ね書きが起こるのを防止するために、前記比較を利用する過程とから成る方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明はコンピュータメモリの分野に関し、特に、重ね書きの問題なくメモリ動作を有利に重複させる装置及び方法に関する。

【0002】

【従来の技術】 データ処理システムは、通常、メモリにデータを記憶し且つメモリのデータを検索する。データの記憶は一般にメモリ中の特定の1つのアドレスを指向する書き込み動作の使用によって実行され、プロセッサからメモリへのデータの供給を含む。データの検索は一般にメモリ中の特定の1つのアドレスを指向する読取り動作の使用によって実行され、メモリからプロセッサへのデータの供給を含む。

【0003】 読取り動作と書き込み動作をプロセッサと、メモリとの間に配置されるメモリ制御装置により管理できると有利である。そのようなメモリ制御装置は、さらに、データ「パイプライン」と呼ばれるものを構成することができる。詳細に言えば、読取りデータをメモリからプロセッサへ供給してゆく読取りパイプラインが書き込みデータをプロセッサからメモリへ供給してゆく書き込みパイプラインとは別になるように、別々の読取りパイプラインと、書き込みパイプラインとを構成することができ

る。別々の読取りパイプラインと書き込みパイプラインを有するメモリ制御装置は「書き込み重複」を可能にし、その場合、書き込み動作は先行する読取り動作が終結する前に始まる。書き込み重複においては、書き込み動作は先行する読取り動作の後のほうの段階で、読取りデータが別個の読取りパイプラインを通過して依然として進行しているときに、メモリ制御装置により開始される。書き込み重複を使用すると、データ処理システムの全体速度は増すので、書き込み重複を使用することが望ましい。

【0004】 ところが、書き込み重複を無制限に使用していると、重複している書き込み動作が先行する読取り動作と同じメモリアドレスを指向しており、先行する読取り動作の中の修正可能なビット誤りがそのメモリアドレスへのライトバック動作が必要となときに問題が生じる。このような状況の下では、不都合なことに重複する書き込み動作がメモリアドレスにデータを書込んだ後に、ライトバック動作が起こる。従って、ライトバック動作により、新たに書き込まれたデータの重ね書きが不適正に起こる。

【0005】

【発明が解決しようとする課題】 以下に説明する通り、本発明は重ね書きの問題を防止する一方で書き込み重複を有利に実行させることを課題とする。

【0006】

【課題を解決するための手段】 本発明は、プロセッサと、メモリと、メモリ制御装置とを有するデータ処理システムの中で動作する。プロセッサとメモリとの間に結合するメモリ制御装置は、メモリから読取られるデータと、メモリに書き込まれるデータとについて別々のデータパイプラインを構成する。一般的にはそれによりメモリ制御装置は書き込み動作を読取り動作と重複させる。さらに、メモリ制御装置は、メモリから読取られているデータの中に修正可能な誤りが検出された場合に、ライトバック動作を実行させる。

【0007】 本発明の現時点で好ましい実施例は、いくつかの書き込み動作を遅延させることにより「重ね書き」の問題を防止する。まず、読取り動作が指向しているメモリアドレスを比較のために記憶する。次に、このメモリアドレスを後続する書き込み動作を指向するメモリアドレスと比較する。2つのアドレスが等価であれば、書き込み待ちカウンタに所定の第1の値をロードし、クロックサイクルごとにカウンタを減分する。後続する書き込みの動作は、書き込み待ちカウンタが所定の第2の値に達するまで開始されない。これにより、不都合な「重ね書き」の発生を防止するために、後続する書き込み動作を重複させるのではなく、所定の長さの時間だけ遅延させるのである。本発明の代替実施例は、読取り動作と、後続する書き込み動作とが同一のアドレスに対するものであるとき、それらの書き込み動作を遅延させるのではなく、読取り動作に関わるライトバック動作を抑制することによ

り、「重ね書き」の問題を防止する。従って、本発明は「重ね書き」の問題を防止する一方で、書き込み重複及びライトバックを実行させる。以下、添付の図面に示す実施例を参照してさらに詳細を説明する。

【0008】

【実施例】データ処理システムで使用するために適用できるメモリ制御装置及びその方法を開示する。詳細に言えば、本発明は、1990年7月17日出願の同時係属出願第07/554,283号に記載されているデータ処理システムに適用可能である。この出願は、デジタルコンピュータによりデータ及びプログラムを記憶し且つ検索するために使用することを特定の適用用途とするダイナミックランダムアクセスメモリ(DRAM)を採用する改良されたシングルインラインメモリモジュール(SIMM)を開示している。本発明の少なくとも一部はこの特定のデータ処理システムに関連して説明されるが、本発明を多様なデータ処理システムで使用しうるとは当業者には理解されるであろう。

【0009】以下の説明中、説明の便宜上、本発明を完全に理解させるために特定のメモリサイズ、データ経路などの詳細な事項を数多く挙げるが、本発明を実施するに際してそのような特定の詳細な事項が要求されないことは当業者には明白であろう。また、場合によっては、本発明を無用にわかりにくくしないために、周知の電氣的構造や回路をブロック線図の形で示すことがあ

る。

【0010】図1は、本発明の教示を利用できると考えられるデータ処理システムの1つを示す。プロセッサ10は、入出力装置14及びメモリ制御装置16を含む様々なシステム構成要素と通信するために、システムバス12に結合している。プロセッサ10は、シングルインラインメモリモジュール(SIMM)20,21,22及び23にあるプログラム及び/又は英数字データやその他のデータから構成されるデータを記憶し且つ検索する。これらのSIMM20~23は、それぞれ、16個のダイナミックランダムアクセスメモリ(DRAM)を含む。図示する通り、SIMM20~23はメモリバス24を介してメモリ制御装置16と通信する。加えて、クロック26は時限デジタルクロック信号をメモリバス24を介してSIMM20~23に供給する。図1はメモリバス24に結合する4つのシングルインラインメモリモジュールを示しているが、本発明をどのような数のSIMMを含むデータ処理システムにも使用しうるとは当業者には理解されるであろう。

【0011】動作中、プロセッサ10はシステムバス12を介して読取り指令と書き込み指令を発行し、システムバス12はそれらの指令をメモリ制御装置16に結合する。読取り動作では、SIMMからメモリバス24を介してメモリ制御装置16にデータに結合し、続いてメモリ制御装置16はデータをシステムバス12に結合し

て、読取る。書き込み動作のときには、データをシステムバス12を介してメモリ制御装置16に結合し、そこで、メモリ制御装置16はデータをメモリバス24を介してSIMMに結合する。データの記憶、データの検索及びSIMM内部のDRAMの復元を実行するために、メモリ制御装置16により様々な制御信号を供給する。詳細に言えば、メモリ制御装置16はSIMMの中に配設されているDRAMに関する行アドレスストロブ(RAS)、列アドレスストロブ(CAS)及びロードパルス(LD_L)と、SIMMのデータの記憶及び検索と関連する他のタイミング信号とを供給する。

【0012】次に、本発明の重ね書き防止システムを伴う書き込みの重複について説明する。図2は、本発明に関連するメモリ制御装置16におけるモジュール及び待ち行列の機能ブロック線図を示す。一般に、システムバス12はシステムバス制御論理30を介してメモリ制御装置16に結合している。詳細に言えば、システムバス12に結合されたメモリ動作指令はシステムバス制御論理30を介してメモリ制御装置16に結合することになる。システムバス制御論理30はメモリ動作指令をメモリ動作指令待ち行列(CMDQUE)32に結合する。たとえば、書き込み動作において、システムバス12に結合されたデータは同様にシステムバス制御論理30を介してデーターイン待ち行列(DIQUE)50に結合する。システムバス12に結合されたメモリアドレスはシステムバス制御論理30を介してアドレス待ち行列(ADRQUE)60に結合する。たとえば、読取り動作において、メモリバス24に結合されたデータはデーターアウト待ち行列(DOQUE)52に結合し、そこで、データーアウト待ち行列はシステムバス制御論理30を介してデータをシステムバス12に結合する。

【0013】メモリ制御装置16はメモリマスターモジュール34と、メモリスレーブモジュール38と、メモリ制御モジュール36とをさらに含む。メモリマスターモジュール34はステートマシンを具現化しており、メモリ動作を実行するためのマスター制御装置である(以下に説明するように、読取り動作はメモリスレーブモジュール38により監視される)。メモリマスターモジュール34はメモリ動作指令待ち行列32に結合しており、この待ち行列32はメモリマスターモジュール34にメモリ動作指令を供給する。メモリマスターモジュール34は書き込み待ちカウンタ(WWC)35を含む。以下に説明するように、書き込み待ちカウンタ35はいくつかの書き込み動作を進行させる前に指定の時間だけ選択的に待たせる。

【0014】ステートマシンを具現化したメモリスレーブモジュール38は、読取り動作を監視し且つ制御する論理を含む。読取り動作と、後続するメモリ動作との重複を可能にするために、この論理はメモリマスターモジュール34とは分離している。メモリスレーブモジュール

ル38は、データアウト待ち行列(DOQUE)52へ制御信号を発行することにより読取り動作のコースを制御する。メモリスレーブモジュール38の主要なタスクは、誤り検出及び修正モジュール70により検出したメモリ誤りを処理するということである。誤り検出及び修正モジュール70はメモリから読取っているデータをメモリバス24を介して受信する。以下に説明するように、誤り検出及び修正モジュール70は、誤りが起こったと判定した場合、誤り制御信号をメモリスレーブモジュール38に結合する。

【0015】メモリマスターモジュール34とメモリスレーブモジュール38は共にメモリ制御モジュールに結合している。メモリ制御モジュール36はタイミング信号RAS、CAS及びLD_Lをメモリバス24を介してSIMMのDRAMに供給する(図1を参照)。加えて、メモリ制御モジュール36はメモリマスターモジュール34と、メモリスレーブモジュール38とにタイミング信号を供給することにより、それらのモジュールにおけるステートマシンのタイミングを制御する。

【0016】次に、本発明の全般的な動作を説明する。読取り動作の場合、読取り指令をシステムバス12からシステムバス制御論理30を介してメモリ動作指令待ち行列(CMDQUE)32に結合する。読取るべき特定のメモリアドレスをシステムバス12からシステムバス制御論理30を介してアドレス待ち行列(ADRQUE)60に結合する。以下に説明する通り、読取り動作では、この特定のメモリアドレスを先行アドレスレジスタ(PAR)80の中に捕獲して、直後に続く書込み動作のメモリアドレスと比較する。メモリ動作指令待ち行列32は読取り指令をメモリマスターモジュール34に結合し、メモリマスターモジュール34は、それに

30 結合して、制御信号をメモリ制御モジュール36に結合することにより、この読取り動作のために必要なタイミング信号を起動する。アドレス待ち行列60は読取り動作に関わるメモリアドレスをメモリ制御モジュール36に供給する。

【0017】次に、メモリから読取るべきデータをメモリバス24へ出力する。データはメモリバス24から誤り検出及び修正モジュール70に結合する。このモジュール70は単一ビット誤り及び二重ビット誤りを検出する。誤りが検出されないと仮定すると、その後、データはデータアウト待ち行列(DOQUE)52に結合し、そこで待ち行列52はデータをシステムバス12に結合する。単一ビット誤り又は二重ビット誤りのいずれかが検出されたと仮定すると、誤り検出及び修正モジュール70はSBE制御信号又はDBE制御信号のいずれかをメモリスレーブモジュール38に供給する。単一ビット誤りの場合のように、ビット誤りが修正可能である場合には、データを修正し、ライトバック動作を開始する。詳細に言えば、メモリスレーブモジュール38はメ

モリマスターモジュール34にSBEW制御信号を供給して、ライトバックを要求する。それに応答して、メモリマスターモジュール34はメモリ制御モジュール36に制御信号を供給し、ライトバックのために必要なタイミング信号を起動する。次に、ライトバックすべきデータをメモリスレーブモジュール38と、メモリ制御モジュール36からのタイミング信号との制御の下にデータアウト待ち行列52からマルチプレクサ72を介してメモリバス24に結合する。

10 【0018】書込み動作の場合には、書込み指令をシステムバス12からシステムバス制御論理30を介してメモリ動作指令待ち行列(CMDQUE)32に結合する。書込みを実行すべき特定のメモリアドレスをシステムバス12からシステムバス制御論理30を介してアドレス待ち行列(ADRQUE)60に結合する。メモリ動作指令待ち行列32は書込み指令をメモリマスターモジュール34に結合し、それに

20 結合してメモリマスターモジュール34は制御信号をメモリ制御モジュール36に結合し、書込み動作に必要とされるタイミング信号を起動する。アドレス待ち行列60は、書込みを実行すべきメモリアドレスをもメモリ制御モジュール36に結合する。メモリに書込むべきデータをシステムバス12からデータイン待ち行列(DIQUE)50に結合する。次に、そのデータをマルチプレクサ72を介してメモリバス24に結合する。

【0019】以上の説明からわかるように、データは本質的にはメモリ制御装置16の中の2つのパイプラインを通して流れてゆく。メモリから読取るべきデータはメモリバス24から誤り検出及び修正モジュール70と、データアウト待ち行列52とを介してシステムバス12へと流れる。メモリに書込むべきデータはシステムバス12からデータイン待ち行列50と、マルチプレクサ54とを介してメモリバス24へと流れる。2つの明確に異なるデータパイプラインを設けることにより、メモリ制御装置16はメモリ動作のある程度の重複を可能にする。詳細に言えば、1つの書込み動作を先行する1つの読取り動作と重複させることができる。たとえば、読取りデータの最後の語がメモリバス24を通り過ぎて、メモリ制御装置16に結合したならば、メモリマスターモジュール34により書込み動作を開始させ、書込みパイプラインを利用することができ、有利である。このようにして、読取り動作の後に書込み動作が続いているときには、書込み動作の早いほうのサイクルは読取り動作の後のほうのサイクルと重複することがありうる。この書込み重複には、データ処理システムの全体速度が増すという利点がある。

【0020】ところが、先に説明した通り、書込み重複を無制限に使用すると、重複する書込み動作が先行する読取り動作と同一のメモリアドレスを指向しており、先行する読取り動作の中に修正可能なビット誤りが必然的

にその特定のメモリアドレスへのライトバック動作を引き起こすときに、問題が生じる。このような状況の下では、重複する書込み動作がメモリアドレスにデータを書込んだ後にライトバック動作が不適正に起こり、その結果、ライトバック動作は新たに書込まれたデータの上に不都合に重ね書きしてしまう。

【0021】本発明の現時点で好ましい実施例は、読取り動作のメモリアドレスを先行アドレスレジスタ(PAR)80に捕獲することにより、この重ね書きの問題を軽減する。読取り動作の後に書込み動作が続いているとき、書込み動作に関わるメモリアドレスを先行アドレスレジスタ80に記憶されている先行する読取り動作のメモリアドレスと比較する。この比較を実行するのは、比較器82のような比較器である。メモリアドレスが等価である場合には、比較器82はADR_EQ制御信号をメモリマスターモジュール34に結合する。それに応答して、メモリマスターモジュール34は書込み動作の開始を、通常、先行する読取り動作を完了するらめに必要な時間の量に対応する期間だけ遅延させる。

【0022】さらに詳細に言えば、ADR_EQ制御信号に応答して、メモリマスターモジュール34は所定の第1の値を書込み待ちカウンタ35にロードする。その後、所定の第2の値に達するまで、クロックサイクルごとに書込み待ちカウンタ35を減分してゆく。メモリマスターモジュール34は、書込み待ちカウンタ35がこの所定の第2の値に達するまで、後続する書込み動作を開始しない。所定の第1の値は、一般に、書込み待ちカウンタがこの所定の第1の値から所定の第2の値まで減分するのに要する時間の量が読取り動作を完了するために必要とされる時間の量に相当するように選択される。そこで、書込み待ちカウンタ35は後続する書込み動作の開始を遅延させて、起こりうる不適正な重ね書きを妨げる。

【0023】現時点で好ましい実施例では、書込み待ちカウンタ35は所定の第1の値から所定の第2の値である0までカウントダウンする。ところが、本発明はこの特定の第2の値には全く限定されないことがわかるであろう。さらに、読取り動作に続く相当の数の書込み動作が同一のメモリアドレスに対するものとは限らないことは明かであろう。従って、本発明はごく限られた数の書込み動作を遅延しつつ相当の書込み重複を可能にする。

【0024】本発明の代替実施例は、後に同じアドレスに向けた書込み動作が続いている読取り動作に関する誤りライトバック動作を抑制する。好ましい実施例におけるのと同じように、読取り動作の後に書込み動作が続いているとき、書込み動作に関するメモリアドレスを先行アドレスレジスタ80に記憶されている先行する読取り動作に関するメモリアドレスと比較する。メモリアドレスが等価である場合には、比較器64はADR_EQ制御信号をメモリマスターモジュール34に結合する。し

かしながら、この代替実施例においては、書込み待ちカウンタを使用して後続する書込み動作をある時間だけ遅延させるのではなく、メモリマスターモジュール34は、先行する読取り動作に関して要求される誤りライトバックを抑制するような論理を含む。先に述べた通り、修正可能な誤りが発見されたときには、メモリスレーブモジュール38はSBEBW制御信号をメモリマスターモジュール34に供給して、誤りライトバックを要求する。この代替実施例では、読取り動作と後続する書込み動作とに関わるメモリアドレスが等価である場合、メモリマスターモジュール34は単にこのライトバック要求を許可しないだけであり、それにより、そのような要求を抑制する。このように、この代替実施例は不都合な重ね書きのおそれをなくすることがわかるであろう。

【0025】図3は、本発明の好ましい実施例の方法を示すフローチャートである。まず、読取り動作のメモリアドレスを比較のために記憶する。後続する書込み動作のメモリアドレスを記憶されている読取り動作のメモリアドレスと比較する。それら2つのアドレスが等価であれば、書込み待ちカウンタに第1の所定の値をロードし、書込み待ちカウンタが第2の所定の値に達するまで、書込み動作を遅延させる。2つのアドレスが等価でない場合には、書込み動作は遅延なく進行して、読取り動作と重複する。

【0026】図4は、先に説明した本発明の代替実施例の方法を示すフローチャートである。まず、読取り動作のメモリアドレスを比較のために記憶する。後続する書込み動作のメモリアドレスを記憶されている読取り動作のメモリアドレスと比較する。それら2つのアドレスが等価であれば、読取り動作に関わる誤りライトバックを抑制し、書込み重複が進行する。2つのアドレスが等価でない場合には、書込み動作は遅延なく進行して読取り動作と重複し、誤りライトバックが許可される。

【0027】本発明を図1～図4を参照して、いくつかのメモリスistemアーキテクチャに重きを置きながら特定して説明したが、図は単なる例示を目的とするだけであり、本発明を限定するものとしてみなされてはならないことを理解すべきである。加えて、本発明の方法及び装置が多数のデータ処理システムのどれにも有用であることは明かである。当業者により、先に開示したような本発明の趣旨から逸脱せずに数多くの変更や変形を実施しようと考えられる。

【図面の簡単な説明】

【図1】本発明の教示を採用するデータ処理システムの1つを示す機能ブロック線図。

【図2】本発明に関連するメモリ制御装置の内部のモジュール及び待ち行列の機能ブロック線図。

【図3】現時点で好ましい実施例の方法を示すフローチャート。

【図4】代替実施例の方法を示すフローチャート。

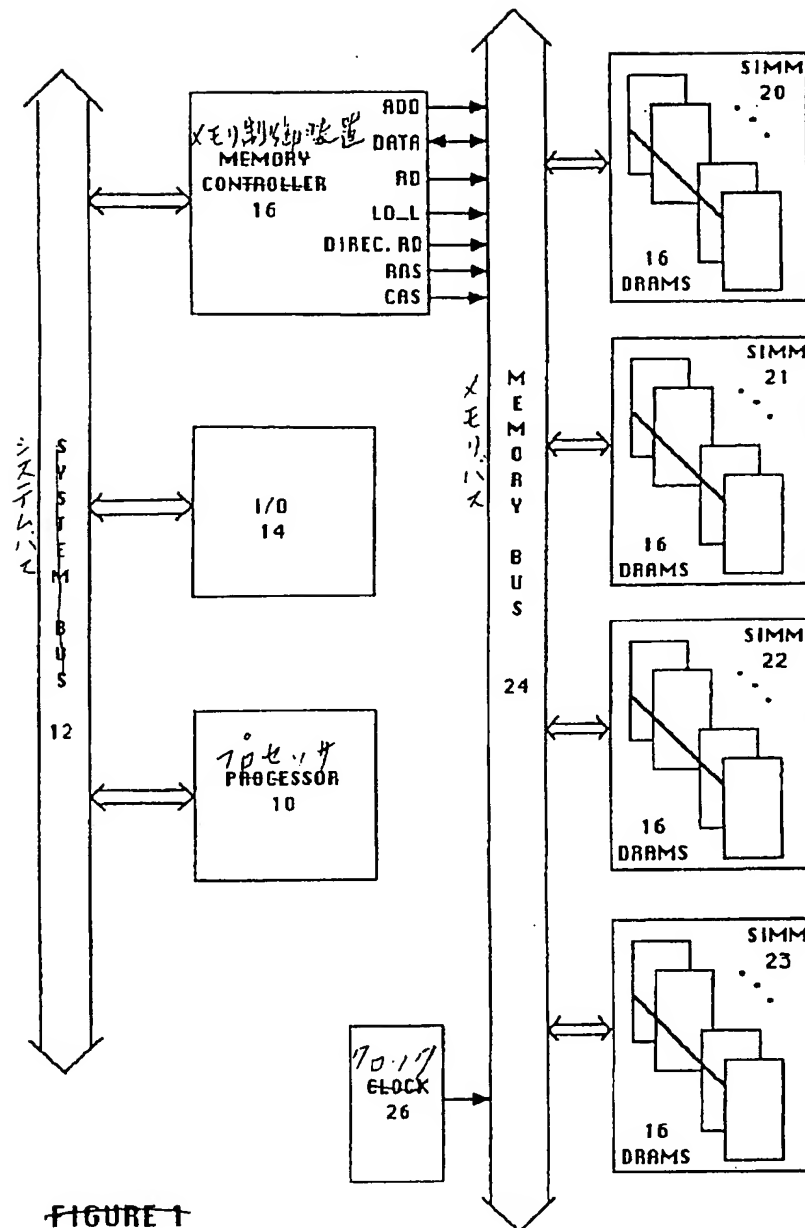
【符号の説明】

10 プロセッサ
12 システムバス
14 入出力装置
16 メモリ制御装置
20～23 シングルインラインメモリモジュール
24 メモリバス
26 クロック
30 システムバス制御装置
32 メモリ動作指令待ち行列
34 メモリマスターモジュール

35 書込み待ちカウンタ
36 メモリ制御モジュール
38 メモリスレーブモジュール
50 データーイン待ち行列
52 データーアウト待ち行列
60 アドレス待ち行列
70 誤り検出及び修正モジュール
72 マルチプレクサ
80 先行アドレスレジスタ
10 82 比較器

【図1】

5433-B



The diagram illustrates the internal architecture of the Memory Controller 16 (メモリ制御装置). It is connected to a **SYSTEM BUS** (12 bits) on the left and a **MEMORY BUS** (24 bits) on the right.

Internal Components and Connections:

- SYSTEM BUS (12 bits):**
 - ADDRQUE 60:** Receives address requests from the system bus and outputs to the **WRITE WAIT-COUNTER 35** and **MEMORY CONTROLLER 36**.
 - DIQUE 50:** Receives data from the system bus and outputs to the **MUX 72**.
 - DOQUE 52:** Receives data from the memory bus and outputs to the system bus.
- Internal Control Logic:**
 - COMMAND 32:** Receives commands from the system bus and outputs to the **MEMORY MASTER 34** and **MEMORY SLAVE 38**.
 - PAR 80:** Receives parity information from the system bus and outputs to the **WRITE WAIT-COUNTER 35** and **MEMORY CONTROLLER 36**.
 - WRITE WAIT-COUNTER 35:** Manages write operations, receiving signals from the **ADDRQUE 60** and **COMMAND 32**, and outputting to the **MEMORY MASTER 34** and **MEMORY CONTROLLER 36**.
 - MEMORY MASTER 34:** Manages memory master operations, receiving signals from the **COMMAND 32** and **WRITE WAIT-COUNTER 35**, and outputting to the **MEMORY SLAVE 38** and **MEMORY CONTROLLER 36**.
 - MEMORY SLAVE 38:** Manages memory slave operations, receiving signals from the **COMMAND 32** and **MEMORY MASTER 34**, and outputting to the **MEMORY CONTROLLER 36**.
 - MEMORY CONTROLLER 36:** Manages memory controller operations, receiving signals from the **ADDRQUE 60**, **WRITE WAIT-COUNTER 35**, and **MEMORY MASTER 34**, and outputting to the **MEMORY BUS** and **EDCM 70**.
- MEMORY BUS (24 bits):**
 - EDCM 70:** Receives data from the memory bus and outputs to the **DOQUE 52**.
 - MUX 72:** Receives data from the system bus and outputs to the **DOQUE 52**.

Handwritten notes in Japanese provide additional context:

- メモリマスター** (Memory Master) is written above the **MEMORY MASTER 34** block.
- メモリスレーブ** (Memory Slave) is written above the **MEMORY SLAVE 38** block.
- メモリ制御装置** (Memory Controller Device) is written below the **MEMORY CONTROLLER 36** block.
- メモリバス** (Memory Bus) is written vertically on the right side of the diagram.
- システムバス** (System Bus) is written vertically on the left side of the diagram.
- 書き込み待ち** (Write Wait) is written near the **WRITE WAIT-COUNTER 35** block.
- ハイフン** (Hyphen) is written near the **MUX 72** block.

FIGURE-2

【図3】

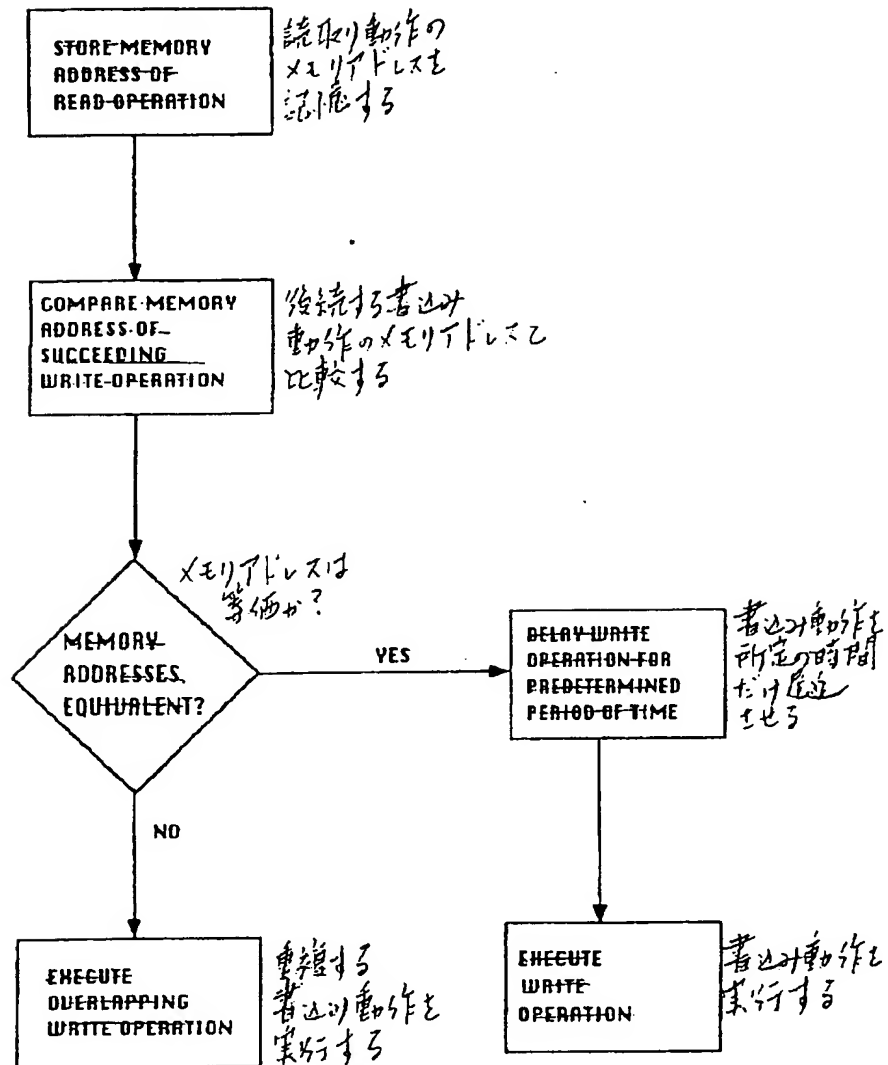


FIGURE 3

【図4】

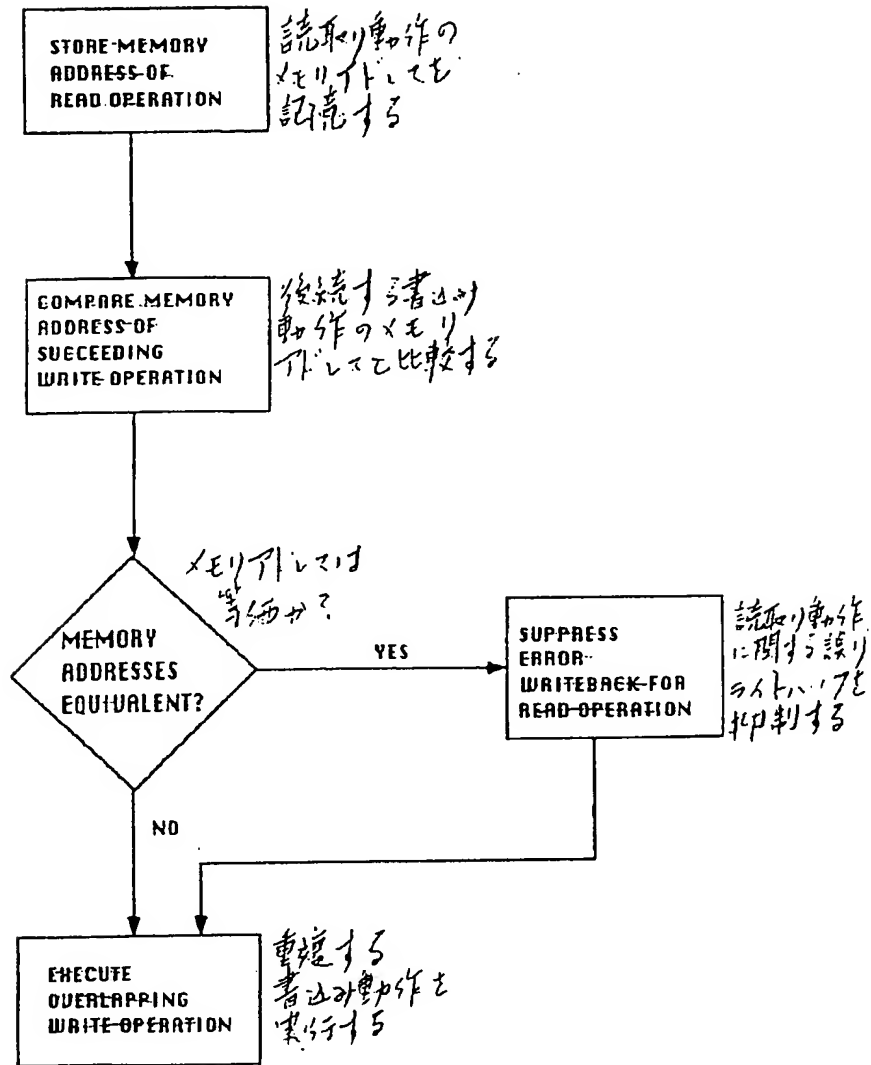


FIGURE 4